مؤسسة بناء وإتصال المعلومات



سلسلة تعليم تصميم وبرمجة تطبيقات أندرويد

جزء أول



إعداد وتأليف

محمد الجيلاني عبد المجيد



موقع الويب : http://www.lbi-egypt.com

بريد إلكتروني : info@lbi-egypt.com

مقدمة الكتاب

بسم الله الرحمن الرحيم والصلاة والسلام على خاتم المرسلين نبينا محمد صلى الله عليه وسلم ، أما بعد ... منذ فترة قريبة قد بدأت أجهزة المحمول التي تعمل بنظام تشغيل أندرويد في الانتشار بشكل واسع على مستوى العالم ، وكل يوم يمر علينا تقوم فيه شركة جوجل بتطوير هذا النظام ونشره نسخه المطورة على شركات صناعة أجهزة المحمول ، بالشكل الذي أدى إلى أن يكون هذا النظام هو النظام الأفضل على مستوى العالم لتشغيل أجهزة المحمول والأجهزة اللوحية في نفس الوقت ، لما له من رونق خاص وسهولة كبيرة في التعامل معه وواجهة رسومية عالية الجوده والمرونة في الأداء ، وليس كل ذلك فقط ، بل وأكثر من ذلك هو سهولة الحصول على تطبيقات كثيرة جداً لأي جهاز يعمل بنظام تشغيل أندرويد . وها نحن إذاً قد وصلنا إلى وجود نوع جديد من أنواع النوافذ التي يتعامل معها المستخدمين على مستوى العالم ، نوافذ تختلف عن نوافذ الكمبيوتر سواء كان محمول أو جهاز مكتبي ، نوافذ يمكن حملها والتنقل بها بدون أي عناء ، لذا فمن واجبنا نحن المبرمجين أن نتعلم كيف يمكننا أن نطور تطبيقات تعمل على هذه النوافذ المتنقلة والمحمولة ، يجب أن ننقل خبراتنا ومعرفتنا إلى هذا المجال وهذه الرقعة الجديدة التي ظهرت على السطح في عالمنا المتطور والمتقدم ، والهدف من هذه السلسلة التعليمة هو دراسة كيف يمكننا تصميم وتطوير تطبيقات تعمل على أجهزة المحمول ذات نظام التشغيل أندرويد . خطوة بخطوة سوف نصل إلى ما نريده بإذن الله ، وتذكر دائماً أنه يجب عليك أن تصمم على الشئ لكي تحصل عليه

مطور تطبيقات

محمد الجيلاني عبد المجيد

محتوى الكتاب

الفصل			الصفحة
الفصل الأول	:	التعريف بنظام تشغيل أندرويد	٤
الفصل الثاني	:	الأدوات اللازمة لبدء العمل وكيفية تشغيلها	٧
الفصل الثالث	:	طريقة عمل وإسلوب برمجة الأنشطة والبرمجيات الفرعية	١٥
الفصل الرابع	:	كيفية استدعاء وربط البرمجيات الفرعية في التطبيق	
الفصل الخامس	:	دراسة محتويات صفحة تطبيقات أندرويد	
الفصل السادس	:	ضبط تصميم واجهة المستخدم مع أبعاد شاشات الأجهزة	
الفصل السابع	:	المكونات الأساسية لواجهة المستخدم في تطبيقات أندرويد	
الفصل الثامن	:	المكونات المتقدمة لواجهة المستخدم في تطبيقات أندرويد	
الفصل التاسع	:	كيفية تصميم معرض صور	
الفصل العاشر	:	إنشاء تطبيق للتعامل مع الكاميرا والتقاط الصور	
الفصل الحادي عشر	:	التعامل مع الكاميرا وإدراج الصور بالخلفية أو حفظها على الميموري كارت	
الفصل الثاني عشر	:	التعامل مع البريد الإلكتروني إرسال وإستقبال	
الفصل الثالث عشر	:	التعامل مع ملفات الصوت وتشغيل الميديا بلاير	
الفصل الرابع عشر	:	تصميم القوائم وتحديد خصائصها	

الفصل الأول : التعريف بنظام تشغيل أندرويد

ما هو أندرويد ؟

أندرويد هو نظام تشغيل مبني على نسخة معدلة من نظام تشغيل لينوكس ، أنشئه من البداية شخص يدعى أندرويد في عام ٢٠٠٥ ، ليكون بداية دخول لعام نظم تشغيل الأجهزة المحمولة ، ثم أشترته شركة جوجل وقامت بتطويره بفريق عمل متخصص يعمل لديها ، أرادت جوجل أن يتم التعامل مع أندرويد كنظام تشغيل مجاني ومفتوح المصدر ، وبالتالي، تم إتاحة التعامل مع تطبيقات أندرويد تحت ترخيص أباتشي مفتوحة المصدر، وهو ما يعني أن أي شخص يريد استخدام أندرويد يمكنه ان يفعل ذلك بتحميل نسخة كاملة ومجانية من أندرويد. وعلاوة على ذلك، الشركات المصنعة للأجهزة المحمولة ، يمكنها إضافة ملحقات خاصة بها وتملكها هي فقط لتمييز منتجاتها عن منتجات الشركات الآخرى . هذا مثال بسيط يجعل عملية تطوير تطبيقات أندرويد جذابة للغاية، وبالتالي لا توجد أي مشكلة بالنسبة لشركات صناعة المحمول من حيث أنها يمكنها تمييز منتجاتها عن منتجات غيرها بتصميم تطبيقات خاصة بأجهزتها ، وتشمل هذه الشركات موتورولا وسوني أريكسون وسامسونج ، ويوماً بعد يوم تزداد الشركات المنتجة والمصنعة للمحمول التي تعتمد نظام تشغيل أندرويد كنظام لتشغيل أجهزتها المحمولة .

مزايا نظام تشغيل أندرويد :-

بصرف النظر عن أنه مفتوح المصدر ، ومجاني ، ومتاح للجميع ، يتمتع نظام تشغيل أندرويد بالمزايا التالية .

١ – التخزين : يتعامل نظام تشغيل أندرويد مع النوع SQLite من أنواع قواعد البيانات العلائقية الخفيفة الحجم .
٢ – الإتصالات : يدعم نظام تشغيل أندرويد عدة أنواع من الشبكات اللاسلكية مثل GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth
٣ – المراسلة : يدعم أندرويد النوعين من المراسلة Short Message Service (SMS), Multimedia Messaging Service (MMS)
٤ – مستعرض الويب : يعتمد على مستعرض ويب خاص به مدعوم من جوجل كروم بالإضافة إلى دعمه للجافا سكريبت وتطبيقاتها المختلفة
٥ – دعم الوسائط المتعددة : يدعم عدة أنواع مختلفة من ملفات الملتيميدا (صوت ، فيديو) .
٦ – دعم الأجهزة والملحقات : يدعم أجهزة الاستشعار والكاميرات وشاشات اللمس والتوجيه الآلي .
٧ – شاشات اللمس المتعدد : يدعم شاشات اللمس المتعدد
٨ – تعدد المهام : يدعم التطبيقات متعددة المهام
٩ – الربط : يدعم أندرويد مشاركة الويب عن طريقة الشبكات السلكية أو اللاسلكية

هيكلية بناء نظام تشغيل أندرويد :-

ينقسم نظام تشغيل أندرويد إلى خمسة أقسام داخل أربعة طبقات رئيسية .
١ ـ نواة لينكس -- وهذا هو النواة التي يعمل من خلالها أندرويد . هذه الطبقة تحتوي على كافة برامج تشغيل الجهاز لمكونات الأجهزة المختلفة.
٢ ـ المكتبات -- وهذه تحتوي على كافة التعليمات البرمجية التي توفر الميزات الرئيسية لنظام تشغيل أندرويد ، على سبيل المثال ، مكتبة SQLite توفر دعم قاعدة ودعم التطبيقات بحيث يمكن استخدامهت لتخزين البيانات. مكتبة WebKit توفر وظائف للتصفح على شبكة الإنترنت.
٣ – Android RunTime – هذه المكتبة تتيح المجال للمطورين لكتابة تطبيقات بلغة جافا ، لتصميم برامج وتطبيقات لتعمل على نظام تشغيل أندرويد ، أي أنها تقوم بعمل ترجمة وتحويل من لغة الجافا إلى لغة يستطيع أن يفهمها نظام تشغيل أندرويد .
٤ - Application framework – هو إطار العمل المسئول عن دعم المطورين بالإمكانيات الإضافية التي يتيحها نظام تشغيل أندرويد ، عن طريق تحديث الأدوات المستخدمة بالفعل وإتاحة الفرصة لتحميل الأدوات الجديدة أو المحدثة .
٥ ـ التطبيقات -- في هذه الطبقة العليا، سوف تجد التطبيقات التي تأتي مع الجهاز (مثل الهاتف، اتصالات، المتصفح، الخ) ، فضلا عن التطبيقات التي تقوم بتحميل والتثبيت من Android Market. وتقع هي والتطبيقات التي سوف نقوم بكتابتها في هذه الطبقة.

أجهزة المحمول التي تعمل بنظام تشغيل أندرويد.

تتوافر الأجهزة التي تعمل بنظام تشغيل أندرويد في عدة صور وأشكال ، منها مايلي :-

Smartphones, Tablets, E-reader devices, Netbooks, MP4 players, Internet TVs





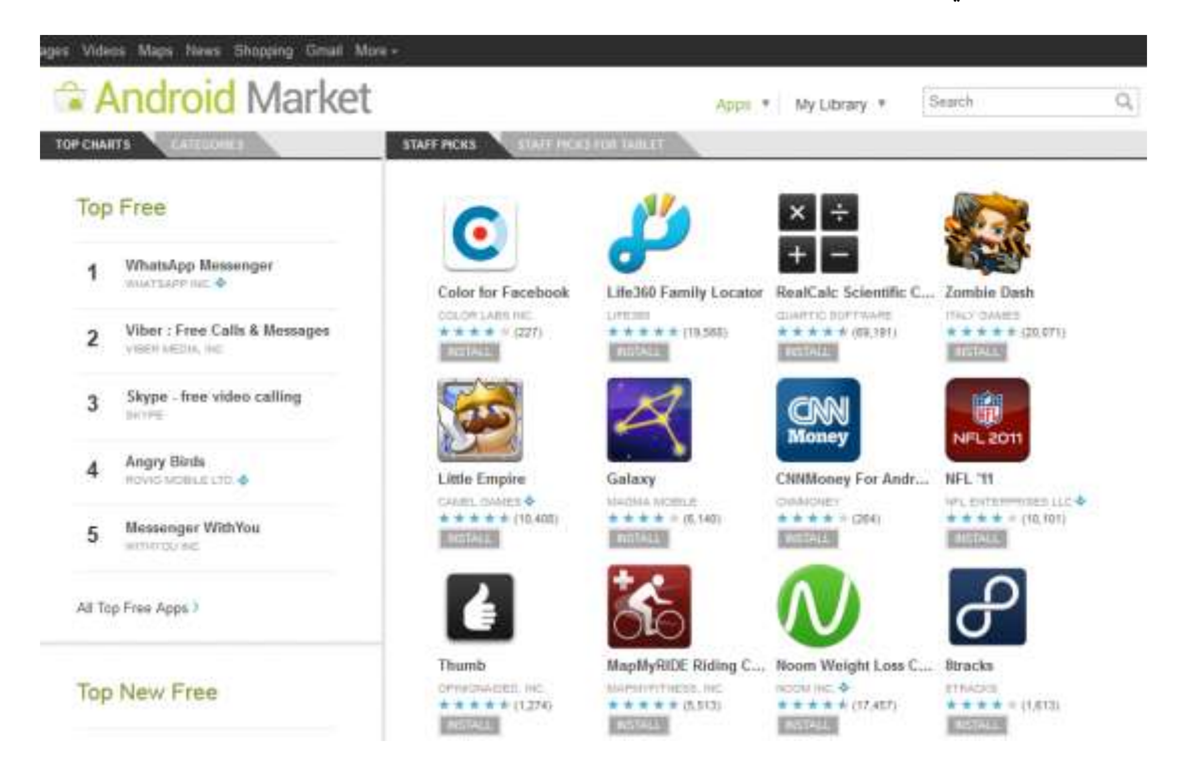






سوق تطبيقات أندرويد Android Market. حالياً يسمى Google Play

كما تحدثنا من قبل ، نظام تشغيل أندرويد مجاني ومتاح للجميع ، وللمزيد من دعم هذه الفكرة قامت شركة جوجل بتصميم وتطوير موقع يتيح تحميل تطبيقات أندرويد للمستخدمين في أي مكان وفي أي وقت ، بحيث أنه يمكن لأي شخص تصميم وبرمجة تطبيق ورفعه على سوق تطبيقات أندرويد ، ويمكنه عرضه مجاناً أو بمبلغ مدفوع ، مما يتيح نشر أكبر عد من التطبيقات للمستخدمين على مستوى العالم تخص جميعها نظام تشغيل أندرويد والأجهزة التي تعمل به .



الفصل الثاني : الأدوات اللازمة لبدء العمل وكيفية تشغيلها

الحصول على الأدوات اللازمة .

للبدء في كتابة أول تطبيق أندرويد يجب أن تمتلك الأدوات اللازمة لذلك ، بدء من جهاز كمبيوتر يعمل بنظام تشغيل لينوكس أو أبل ماكنتوش أو ويندوز ، ويتبقى مجموعة من التطبيقات المجانية التي يمكن تحميلها من الويب بسهولة . وذلك بإتباع الخطوات التالية .

١ – قم بتحميل Java SE Development Kit (JDK) من الرابط التالي

http://www.oracle.com/technetwork/java/javase/downloads/index.html



٢ – الخطوة الثانية قم بتحميل تطبيق البيئة التكاملية للتطوير integrated development environment (IDE) وأفضل تطبيق يخدم أهدافنا هو تطبيق Eclipse ، وهو تطبيق يعمل لديك على الجهاز بدون الحاجة إلى تنصيبه على الجهاز ، فهو يعتبر نسخة محمولة بمجرد تنزيله من الويب وفك ضغطه إلى مجلد معين ، يمكنك تشغيله من هذا المجلد بشكل مباشر ، وعنوان التحميل هو

http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/helios/SR1/eclipse-jee-helios-SR1-win32.zip

٣ – الخطوة الثالثة هي تحميل نسخة نظام تشغيل أندرويد من الرابط التالي

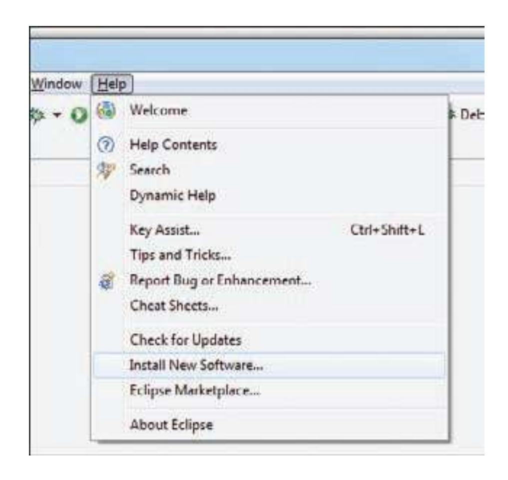
http://dl.google.com/android/installer_r16-windows.exe

بعد تحميله قم بفك بفك الضغط أو تثبيته في المجلد

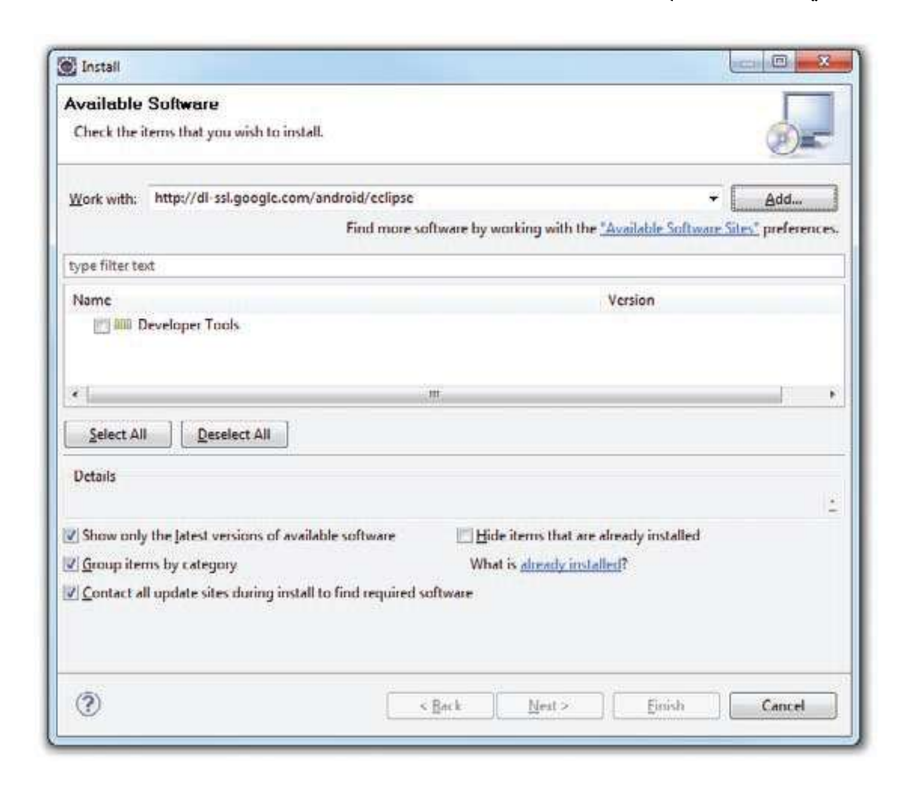
C:\Android\

٤ – يتبقى الأن تحميل أدوات تطوير أندرويد Android Development Tools (ADT) .
وذلك من خلال إتباع الخطوات التالية :-

- قم بتشغيل برنامج Eclipse ثم من قائمة Help إختر Install New Software

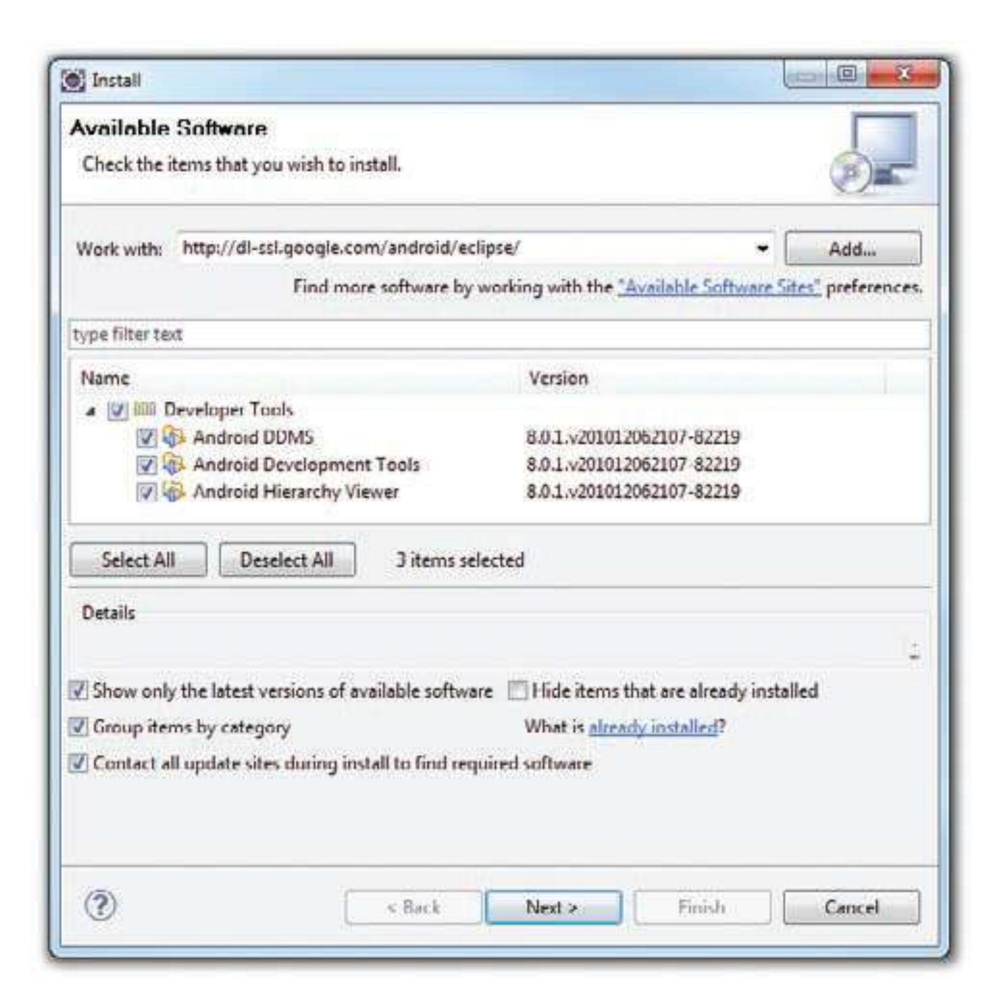


- إضغط على زر Add في الجزء العلوي جهة اليمين

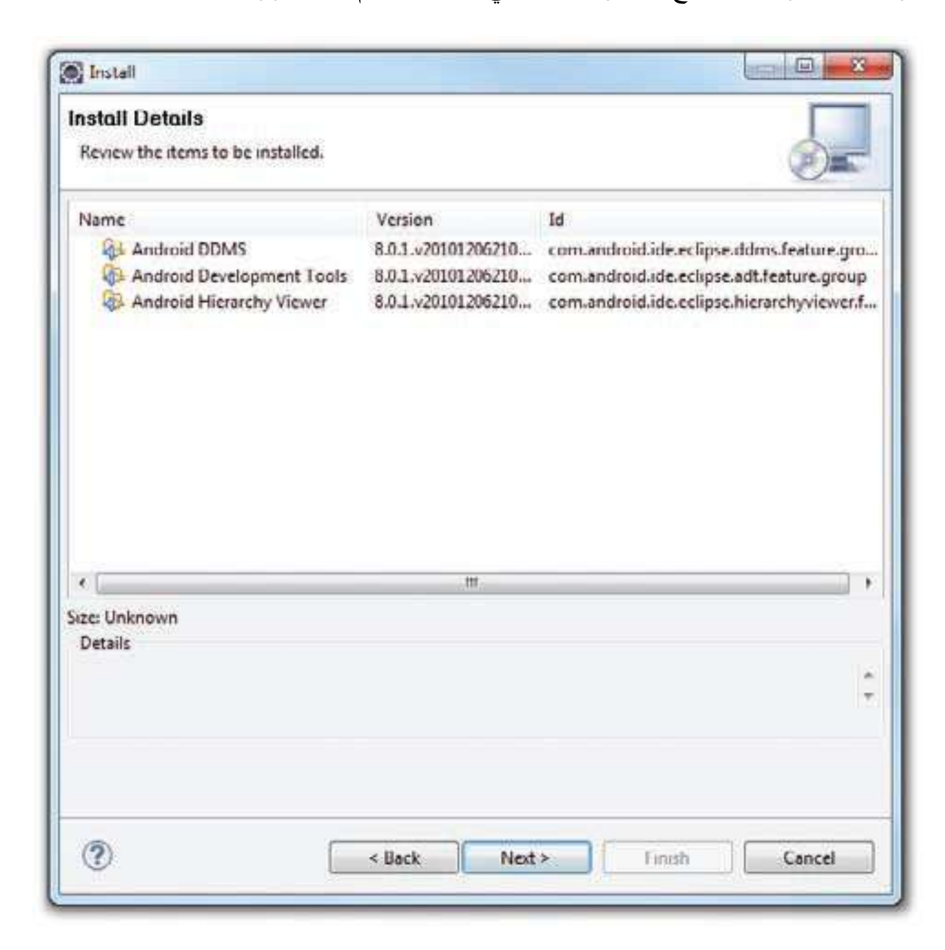


- سوف تظهر لك شاشة فيها خانة للاسم Name أكتب بها ADT Plugin ثم في خانة URL for the Location أكتب العنوان التالي ليتم تحميله للبرنامج https://dl-ssl.google.com/android/eclipse/

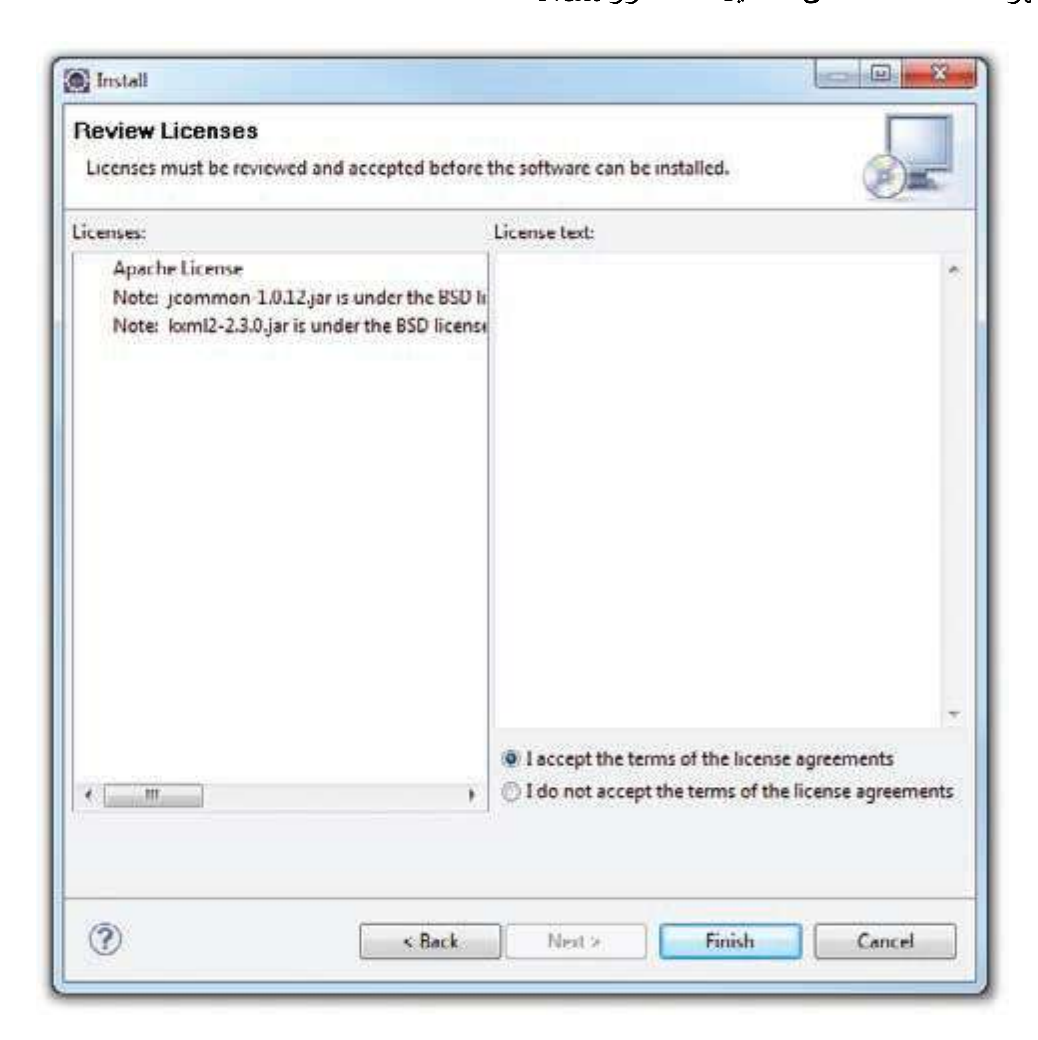
- إضغط زر Ok



- في الخانة التي تظهر أمامك أختر علامة صح لتطوير النسخة التي تعمل عليها ثم أضغط زر Next



- سوف تظهر لك شاشة تسألك عن التحميل أضغط زر Next



- أخيراً إضغط زر Finish .



- أغلق برنامج Eclipse ثم أعد فتحه مرة آخرى .

بعد فتح برنامج Eclipse ، قم بإتباع الخطوات التالية

- من قائمة Window أختر أمر Preferences

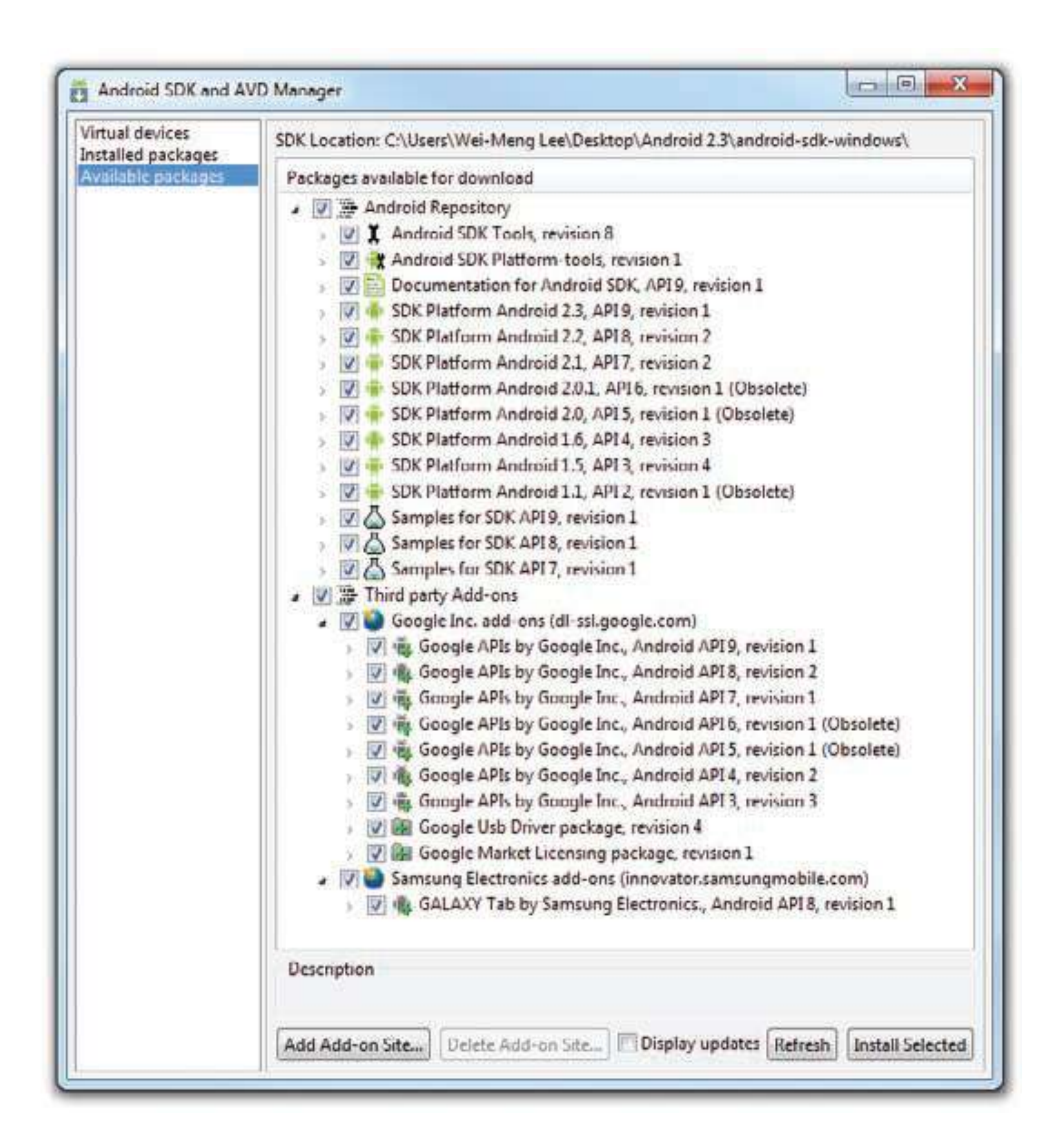
- سوف تظهر أمامك شاشة تحديد من خلالها مكان تواجد مجلد Android SDK الذي قمنا بتحميله من قبل



الأن بعد أن قمنا بتعريف مكان وجود Android SDK ، يتبقى أن نضيف المحرك الافتراضي لتشغيل Android .

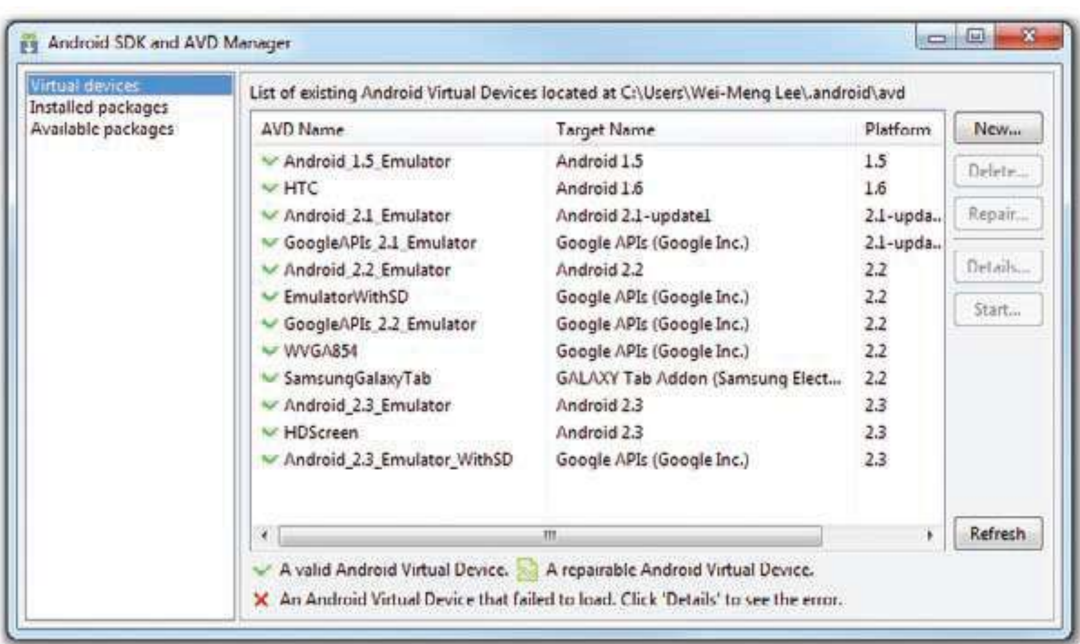
من خلال القيام بالضغط على أمر Android SDK And AVD Manager .

قم باختيار خيار Available packages ثم قم بتحديد الخيارات التي تجدها على الجانب الأيمن ، ثم قم بالضغط على زر Install Selected .



من الشاشة السابقة نختار التبويب Virtual Devices ، ثم نضغط زر New الموجود على الجانب الأيمن لنقوم بإضافة محرك افتراضي لتشغيل نظام تشغيل أندرويد . نقوم بإضافة خانة Name بحيث تعبر عن نسخة نظام التشغيل المحدد من قائمة Target . وهي تعتبر إصدرات مختلفة لنفس نظام التشغيل – مع توضيح إختلاف أبعاد الشاشات

Android SDK and AVD Manager
Virtual devices
Installed packages
Available packages
List of existing Android Virtual Devices located at C:\Users\Wei-Meng Lee\.android\avd
AVD Name
Target Name
Platform
API Level
New...
Delete...
Repair...
Details...
Start...
Refresh
Create new Android Virtual Device (AVD)
Name: Android_2.3_Emulator
Target: Android 2.3 - API Level 9
SD Card:
Size: MiB
File: Browse...
Skin:
Built-in: Default (HVGA)
Resolution: x
Hardware:
Property Value New...
Abstracted LCD density 160 Delete
Override the existing AVD with the same name
Create AVD Cancel



الشكل السابق يوضح إمكانية إضافة عدة محركات افتراضية لنظام تشغيل أندرويد



أسئلة الفصل الثاني

س١ : ما هي الأدوات اللازمة لتطوير تطبيقات أندرويد ؟

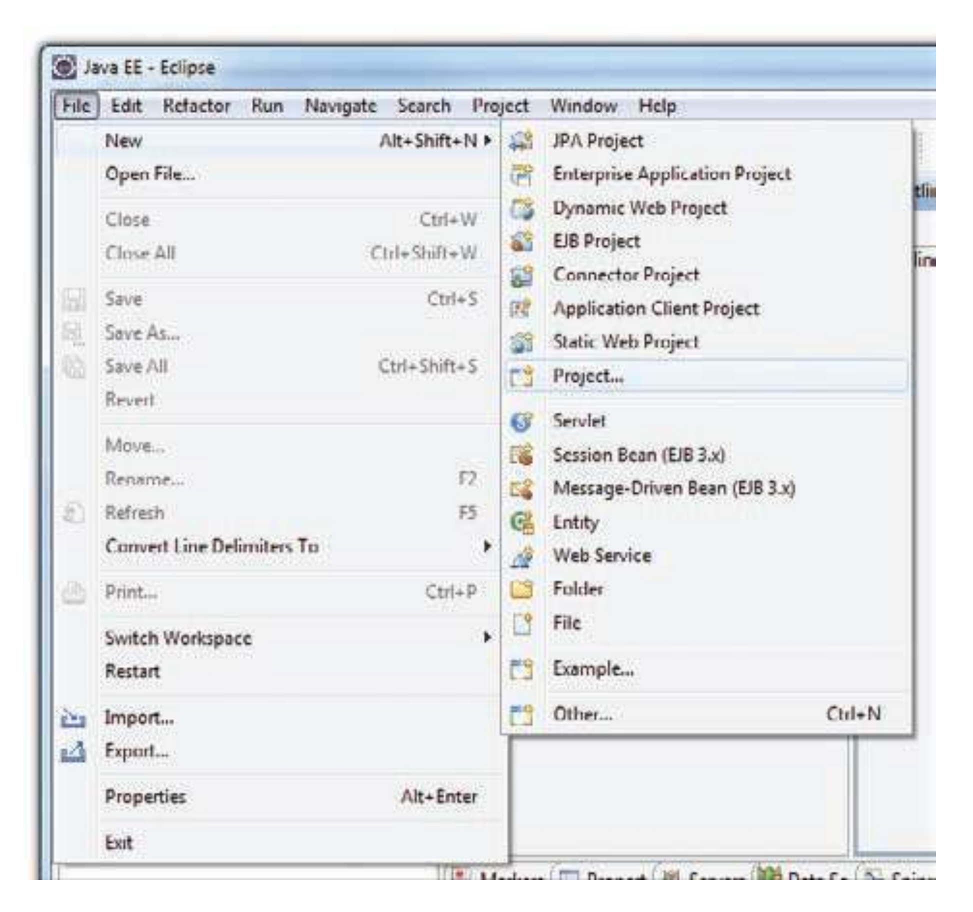
س٢ : ما هي أسعار الأدوات اللازمة لتطوير تطبيقات أندرويد ؟

س٣ : ما هو نظام التشغيل أو نظم التشغيل التي يمكنك من خلالها تطوير تطبيقات أندرويد ؟

الفصل الثالث : طريقة عمل وإسلوب برمجة الأنشطة والبرمجيات الفرعية

لفهم ودراسة طريقة عمل وأسلوب برمجة الأنشطة والبرمجيات الفرعية ، سوف نبدأ معاً في كتابة أول تطبيق تعليمي في الدورة ، وذلك بإتباع الخطوات التالية .

١ – قم بتشغيل برنامج Eclipse ومن قائمة File أختر أمر Project كما هو مبين في الصورة التالية .



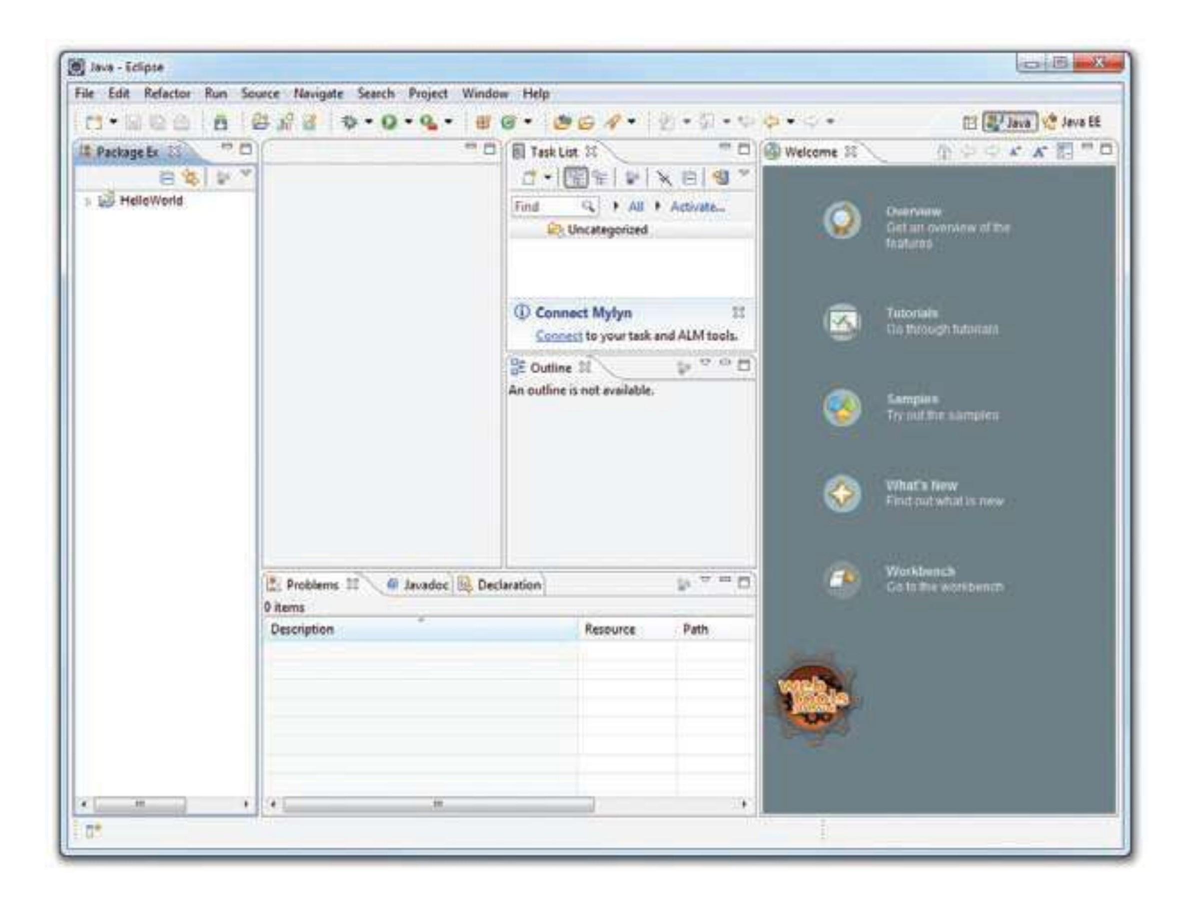
٢ – من النافذة التالية أختر نوع المشروع أنه Android Project .

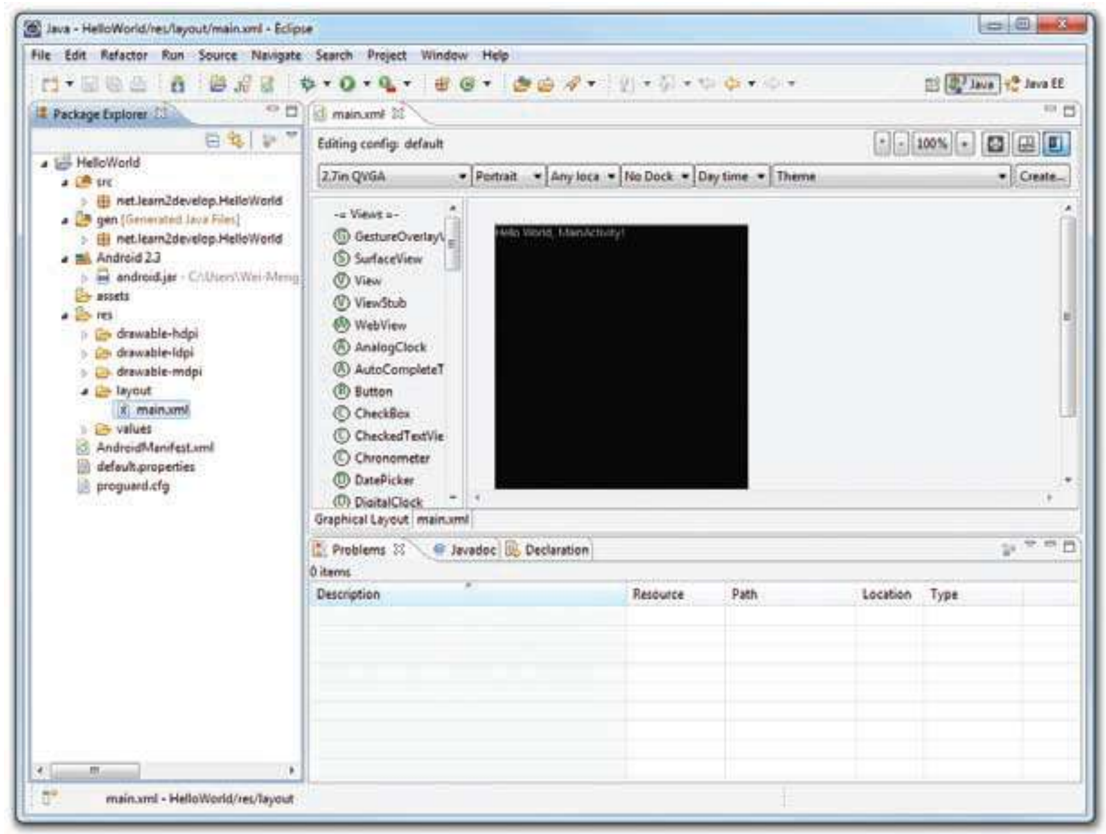


٣ – من النافذة التالية نحدد اسم المشروع الذي سوف نقوم بإنشائه . مع تحديد الإصدار المستهدف كحد أدنى لتشغيل التطبيق .

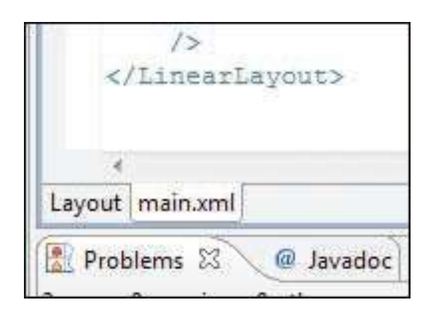


٤ – نضغط على زر Finish ، ومن المفترض أن يكون لدينا الشكل التالي مع بعض التغييرات البسيطة .





نلاحظ في الصورة السابقة أننا نقف في مستعرض ملفات المشروع على الملف المسمى mail.xml ، وهو عبارة عن صفحة في التطبيق ، أو عبارة عن النافذة الرئيسية للتطبيق الذي سوف نقوم بتصميمه ، وإذا ضغطنا على التبويب الموضح في الصورة التالية



سوف نرى الكود التالي مكتوب في صفحة main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="fill_parent" >
<TextView
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="@string/hello" />
<TextView
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="This is my first Android Application!" />
<Button
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="And this is a clickable button!" />
</LinearLayout>
```

لحفظ المشروع قم بالضغط على Ctrl+s ، ولتشغيل المشروع قم بالضغط على F11 ومن الممكن أن يسألك برنامج Eclipse ويظهر لك الشاشة التالية وينتظر منك الإجابة .

ما قمنا بكتابته فقط هو

```
<TextView
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="This is my first Android Application!" />

<Button
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="And this is a clickable button!" />
```

المنطقتين السابقتين هما اللتان كتبناهما فقط ، حيث قمنا بكتابة أول منطقة لإنشاء خانة عرض بيانات جديدة TextViewونلاحظ منها الأتي -:

١ – الخاصية : "android:layout_width = "fill_parent وهي تعني أن قيمة خاصية عرض هذه الخانة هو قيمة العرض الكامل لصفحة الموبايل .

٢ – الخاصية : "android:layout_height="warp_content وهي تعني أن خاصية إرتفاع الخانة يساوي إرتفاع الكتابة التي عليها ، أي أنها ترتفع بمقدار إرتفاع الكتابة سطر أو سطرين أو أكثر بحسب إرتفاع الكتابة فقط .

٣ – الخاصية : "android:text="this is my first android application وهي قيمة النص المكتوب في هذه الخانة ، وبالتالي هو النص الذي يظهر لنا عند تشغيل التطبيق .

وقمنا بتكرار نفس الشئ مع أداة آخرى تسمى Button أي زر أمر ، وقمنا بتحديد نفس الخصائص بالنسبة له .



سوف نجد أن محاكي أجهزة أندرويد تم تحميله وقام بعرض التطبيق الذي قمنا بكتابته بالشكل التالي



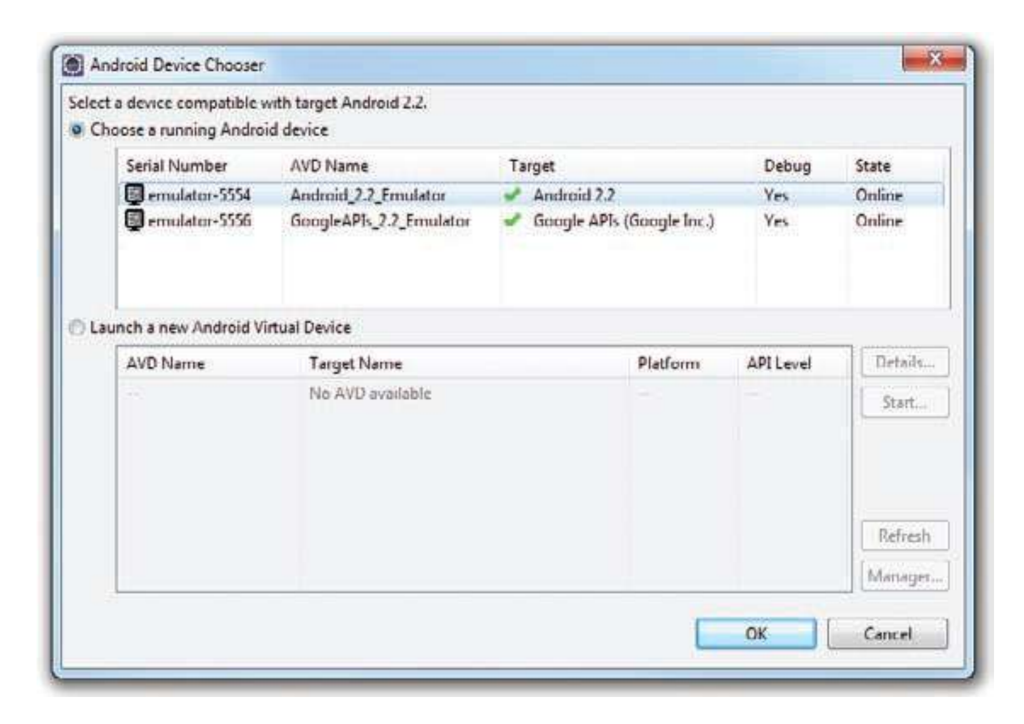
ويمكن بطريقة آخرى تشغيل التطبيق من خلال الخطوات التالية





ملحوظة هامة :-

سبق وأن قمنا بتحديد العديد من الأجهزة الافتراضية ، لذا يسألنا برنامج Eclipse بصفة مستمرة عن الجهاز الذي نود تجربة التطبيق عليه ، ويظهر لنا شاشة توضح لنا كافة الأجهزة التي قمنا بتسجيلها ، كما يمكننا أن نقوم بإضافة أجهزة جديدة .



بصفة عامة ، يجب أن تكون على علم ببعض المسميات المهمة أثناء قيامك بتصميم تطبيق أندرويد ، وفيما يلي نوضح أهم المفاهيم الواجب معرفتها في التطبيق .

الخاصية	الوصف
Project name	خاصية اسم المشروع
Application name	الاسم السهل بالنسبة للمستخدم الذي يطلقه على التطبيق
Package name	اسم المشروع مرتبط مع التطبيق
Create Activity	اسم النشاط الرئيسي في التطبيق (النشاط هو عبارة عن ملف جافا يحتوي على البرمجيات الرئيسية التي يتم تشغيلها في بداية المشروع) .
Min SDK Version	الحد الأدنى من إصدار SDK الذي يحتاجه تطبيقك ليعمل بشكل طبيعي .

كما يجب أن تعلم كل مجلد في تطبيق أندرويد ما هي وظيفته وماذا يحتوي .

المجلد : src يحتوي على ملف الجافا الرئيسي للمشروع ويمكن أن يحتوي على ملفات جافا آخرى إضافية ، وهي يحتوي على السورس كود البرمجي وكافة التطبيقات أو البرمجيات الفرعية التي تنفذ أوامر معينة في المشروع . وهو يحتوي بشكل أساسي على الملف MainActivity.java	
المجلد : Android 2.3 هو مكتبة إصدار أندرويد التي تستخدمها في مشروعك وهو يحتوى بشكل أساسي على ملف android.jar ومن الممكن أن يحتوي على ملفات آخرى لكل ملف عبارة عن مكتبة يتم إستخدامها في التطبيق الخاص بك .	
المجلد gen يحتوي على الملف R.java ويتم توليد هذا الملف من خلال المترجم Compiler الخاص بلغة جافا الموجود ضمن برنامج Eclipse ، ولا يجب عليك مطلقاً تعديل هذا الملف .	
المجلد assets يحتوي على كافة الموارد الخاصة بك التي يمكنك استخدامها في تطبيق أندرويد خاص بك ، مثل ملفات HTML, Text, Database, Image . وهذا المجلد خاص بالموارد التي تخص تطبيقك والتي تعتبر خاصة بكل مشروع على حدى ، أي موارد استثنائية	

.	
المجلد res يحتوي على الموارد الأساسية للتطبيق ، مثل الصفحات ، الصور والأيقونات بأبعادها الثلاثة المختلفة ، القيم التي تخزن لتخصيصها للأدوات المستخدمة في التطبيق .	
الملف AndroidMainfest.xml من خلال هذا الملف يمكنك تخصيص صلاحيات التطبيق ومواصفاته ، وسوف نتحدث عنه بشكل أكثر تفصيلاً في المحاضرات القادمة .	



أسئلة الفصل الثالث

س١ : ما هو AVD ؟

س٢ : ما هي المجلدات التي يحتوي عليها أي مشروع أندرويد ؟

الفصل الرابع : كيفية استدعاء وربط البرمجيات الفرعية في التطبيق

سوف نبدأ بدراسة كيف يتم إنشاء الأنشطة (Activities) ، حيث أنك حينما تقوم بإنشاء نشاط فإنك تقوم بإنشائه مبنياً على أساس ملف النشاط الرئيسي للمشروع الخاص بك ، أي أنه له مرجعية يعتمد عليها في تشغيله ، والمثال التالي يوضح كود نشاط بسيط جداً يتم توليده كلما بدأت مشروع جديد

```
1   package net.learn2develop.Activities;
2   import android.app.Activity;
3   import android.os.Bundle;
4   public class MainActivity extends Activity {
5   /** Called when the activity is first created. */
6   @Override
7   public void onCreate(Bundle savedInstanceState) {
8   super.onCreate(savedInstanceState);
9   setContentView(R.layout.main);
10  }
11  }
```

السطور السابقة تدل على الأتي شرحه .

١ – أمر يستدعي أسم الباقة التي نعمل عليها والخاصة بنفس المشروع محل العمل .
٢ – يتم إستدعاء المكتبة المسماه android.app.activity والمسئولة عن تشغيل هذا النشاط في بدء تشغيل التطبيق .
٣ – يتم إستدعاء مكتبة مسئولة عن تبديل قيم المتغيرات ، وهذه مهمة للتعامل مع القيم المختلفة وتخصيصها للأدوات .
٤ – بداية كود النشاط وتعريفه على أنه نشاط عام public من النوع class له اسم محدد MainActivity مبني على extends المكتبة الرئيسية للأنشطة Activity .
٥ – سطر تعليق لا يلتفت اليه مترجم اللغة ولكن مهم بالنسبة لنا لنعرف أو لنوضح ما هو المقصود من الجملة التالية له .
٦ – أمر يكتبه مولد اللغة ويكتبه قبل كل منطقة كود يتم توليدها تلقائياً .
٧ – إجراء عام يتم تنفيذه في حالة إنشاء التطبيق ، مع إعطاء الأمر بحفظ حالة التطبيق من خلال saveedINstanceState .
٨ – تنفيذ أمر حفظ إعدادات التطبيق على الجهاز .
٩ – عرض الصفحة الرئيسية من خلال الأمر setContentView بحيث يحتوي على R.layout.main بحيث يكون النتيجة هي إظهار الصفحة الرئيسية في التطبيق كأول صفحة عند تشغيل هذا الكود .

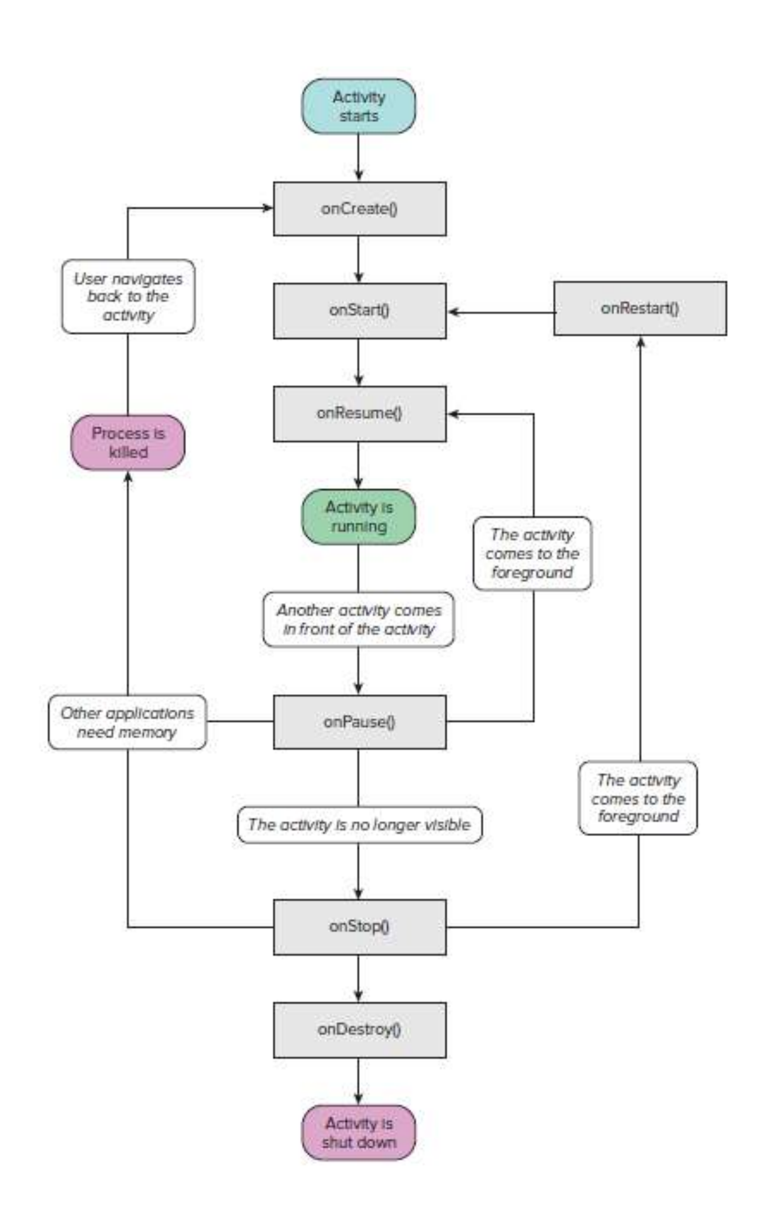
نلاحظ بعد ذلك أنه لابد من تعريف كل نشاط موجود في التطبيق ، ومكان تعريفه هو ملف AndroidMainfest.xml بالشكل التالي .

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="net.learn2develop.Activities"
android:versionCode="1"
android:versionName="1.0">
<application android:icon="@drawable/icon"
android:label="@string/app_name">
<activity android:name=".MainActivity"
android:label="@string/app_name">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category
android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>
<uses-sdk android:minSdkVersion="9" />
</manifest>
```

الأنشطة Activities تتعرض أثناء مراحل تشغيلها لوقوع العديد من الأحداث ، لكل حدث وقت حدوث معين ومحدد ، لذا فمن الممكن أن نكتب لكل حدث أمر معين ينفذ فقط عندما يحدث هذا الحدث . وفيمايلي أوقات حدوث هذه الأحداث المختلفة .

اسم الحدث	وقت حدوثه
onCreate()	يحدث عند تشغيل التطبيق ، حيث يتم بناءه في الذاكرة
onStart()	يحدث حينما يبدء التطبيق في الظهور أمام المستخدم
onResume()	يحدث عندما يتم التفاعل بين المستخدم وبين التطبيق
onPause()	يحدث عند توقف النشاط الحالي مع تشغيل النشاط السابق
onStop()	يحدث عندما لا يتم التعامل مع التطبيق لفترة طويلة
onDestroy()	يحدث عندما يتم إيقاف التطبيق من خلال النظام لتوفير الذاكرة أو يدوياً
onRestart()	يحدث عندما يتم إيقاف تشغيل التطبيق وإعادة فتحه مرة آخرى

بشكل طبيعي يتم تنفيذ الحدث onCreate() كأول حدث في النشاط ، لذا ستجد أن أي نشاط يبدأ بهذا الحدث ، وفيما يلي خريطة توضح لنا كيفية سير الأحداث في مراحل تشغيل التطبيق .



وللمزيد من فهم الأحداث سوف نقوم بتصميم برنامج بسيط يرسل لنا رسائل معينة عند حدوث كل حدث .

١ – قم بإنشاء مشروع جديد واعطه الاسم Activities ، وحدد اسم الباقة بـ net.learn2develop.Activities

قبل ذلك من المهم أن تتبع الخطوات التالية – من قائمة windows اختر Show View ثم اختر Other ثم اختر Android ثم اختر LogCat (deprecated) .

٢ – في ملف ActivitiesActivity قم بتعديل الكود البرمجي ليصبح كما بالمثال التالي .

```
package net.learn2develop.Activities;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;

public class ActivitiesActivity extends Activity {
       String tag = "Events";

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Log.d(tag, "In the onCreate() event");
    }
    public void onStart()
    {
    super.onStart();
    Log.d(tag, "In the onStart() event");
    }
    public void onRestart()
    {
    super.onRestart();
    Log.d(tag, "In the onRestart() event");
    }
    public void onResume()
    {
    super.onResume();
    Log.d(tag, "In the onResume() event");
    }
    public void onPause()
    {
    super.onPause();
    Log.d(tag, "In the onPause() event");
    }
    public void onStop()
    {
    super.onStop();
    Log.d(tag, "In the onStop() event");
    }
    public void onDestroy()
    {
    super.onDestroy();
    Log.d(tag, "In the onDestroy() event");
    }
}
```

إظهار الشاشة الحوارية Dialog Window .

من خلال المثال الحالي سوف نتعلم كيف يمكننا إنشاء نموذج حواري يسأل المستخدم عن أشياء معينة ، بدون أن نقوم بتصميم عناصر هذا النموذج بشكل فعلي على الشاشة ، بمعنى أوضح نريد أن نقوم بتصميم نموذج بسيط يسأل المستخدم عن أحد الأشياء ويستمع لإجابة المستخدم ليعيد للبرنامج إجابة البرنامج على الأسئلة أو السؤال الذي تم طرحه عليه .

١ – قم بفتح برنامج Eclipse وقم بكتابة برنامج جديد باسم Dialog .

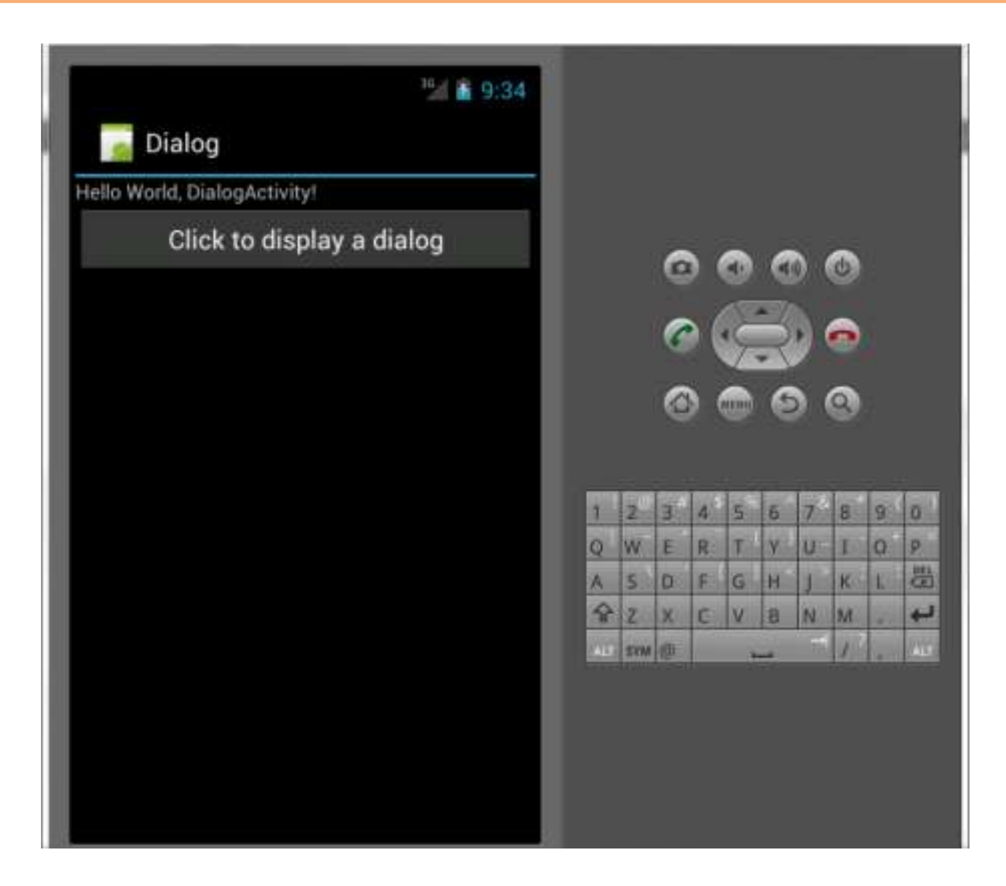
٢ – سوف نقوم بإضافة زر أمر Button إلى ملف main.xml بحيث تصبح الشاشة بها زر أمر جديد ، ويصبح كود الصفحة الرئيسية بهذا الشكل .

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

      <Button
      android:id="@+id/btn_dialog"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="Click to display a dialog" />

</LinearLayout>
```



٣ – قم بكتابة الكود التالي بحيث يصبح ملف DialogActivity.java بالشكل التالي .

```
package net.learn2develop.Dialog;

import android.app.Activity;
import android.os.Bundle;

import android.app.AlertDialog;
import android.app.Dialog;
import android.content.DialogInterface;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class DialogActivity extends Activity {

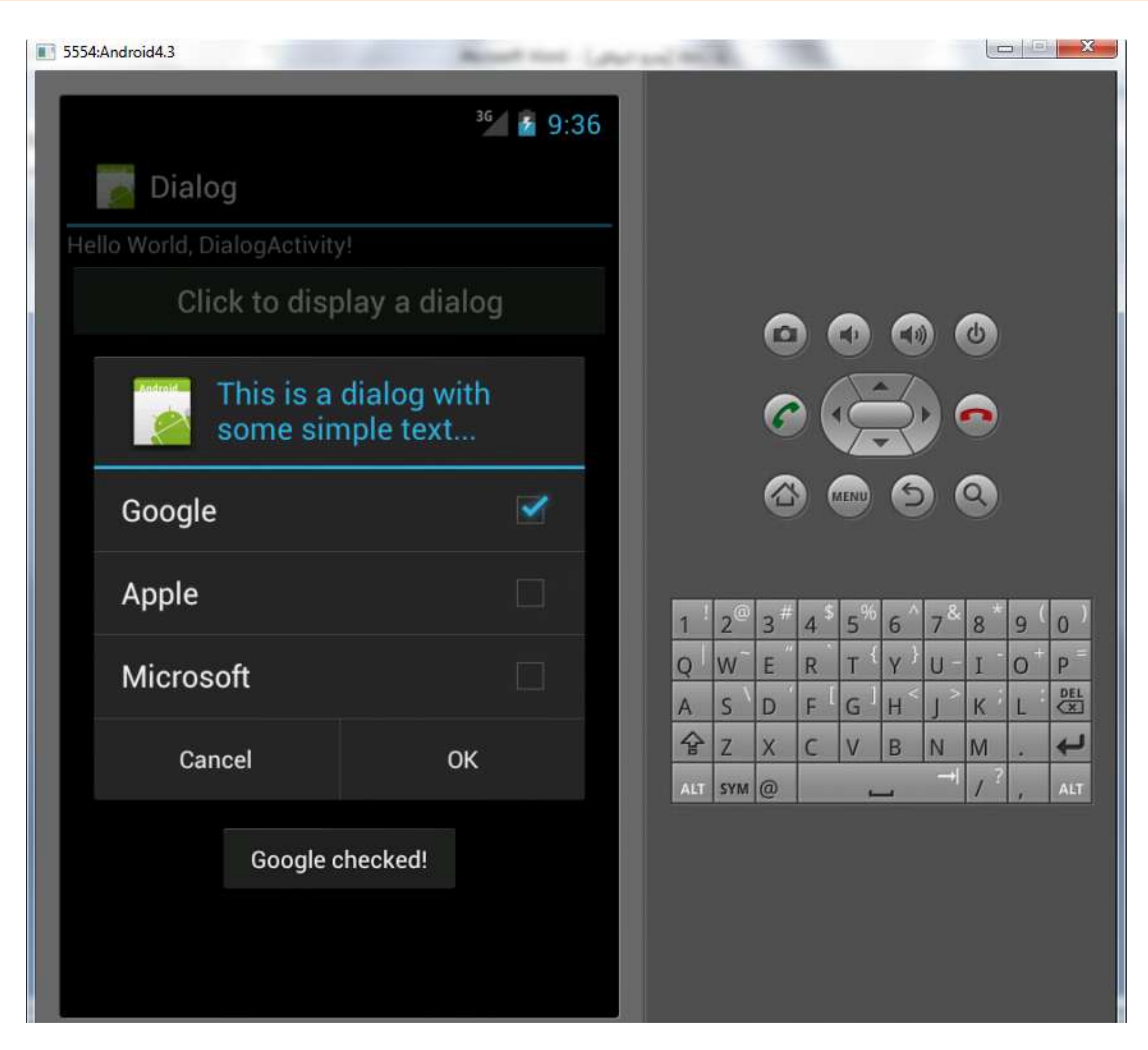
      CharSequence[] items = { "Google", "Apple", "Microsoft" };
      boolean[] itemsChecked = new boolean [items.length];

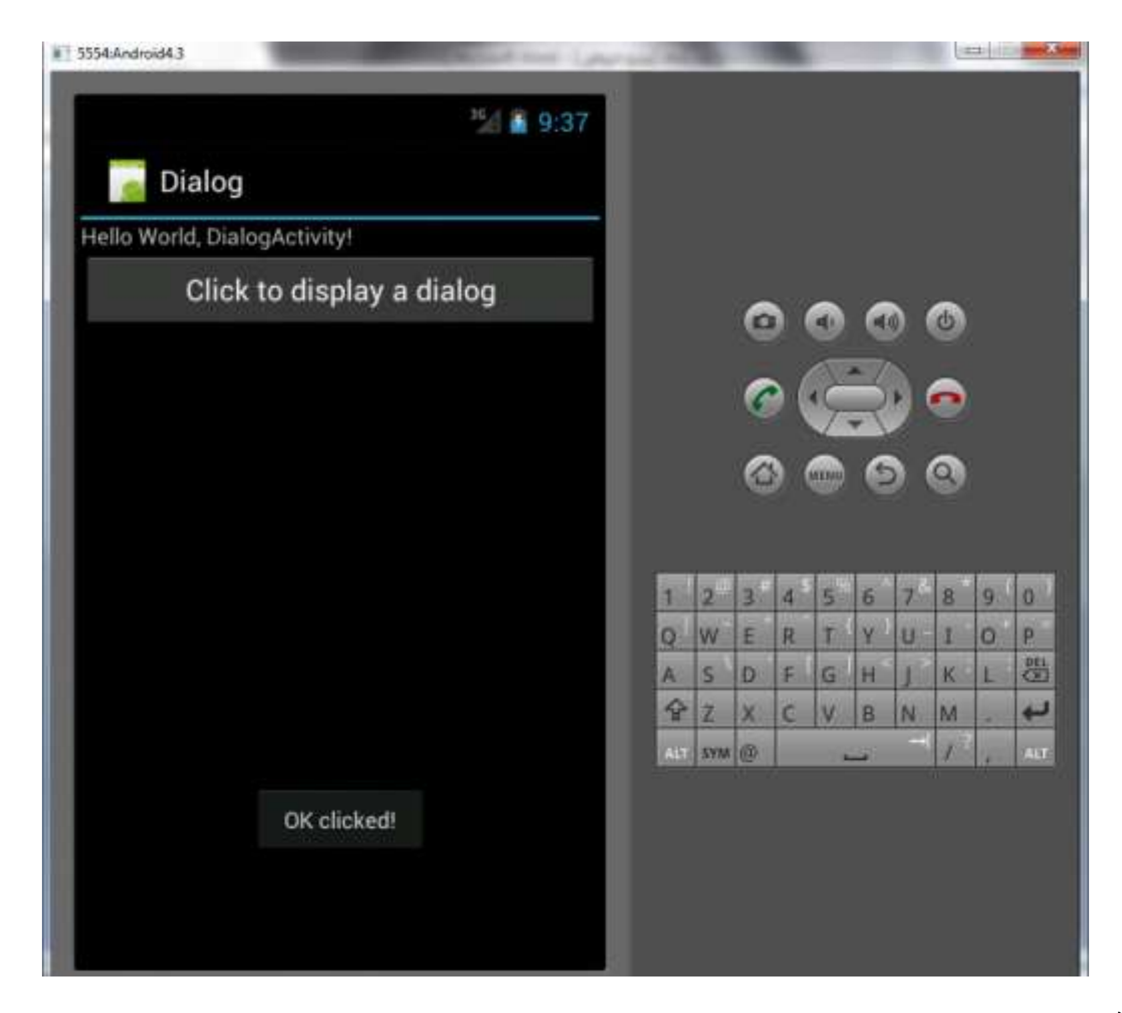
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        Button btn = (Button) findViewById(R.id.btn_dialog);
        btn.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
        showDialog(0);
        }
        });
    }

    @Override
    protected Dialog onCreateDialog(int id) {
          switch (id) {
          case 0:
          return new AlertDialog.Builder(this)
          .setIcon(R.drawable.ic_launcher)
          .setTitle("This is a dialog with some simple text...")
          .setPositiveButton("OK", new
          DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog,
                int whichButton)
          {
                  Toast.makeText(getBaseContext(),"OK clicked!",
Toast.LENGTH_SHORT).show();
          }
          })
          .setNegativeButton("Cancel", new
          DialogInterface.OnClickListener() {
          public void onClick(DialogInterface dialog,
          int whichButton)
          {
            Toast.makeText(getBaseContext(),"Cancel clicked!",
Toast.LENGTH_SHORT).show();
          }
          })
          .setMultiChoiceItems(items, itemsChecked, new
          DialogInterface.OnMultiChoiceClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int
which,
                boolean isChecked) {
```

```
                                        Toast.makeText(getBaseContext(),
                                        items[which] + (isChecked ? " checked!":
                                        " unchecked!"),
                                        Toast.LENGTH_SHORT).show();
                                        }
                                }
                                )
                                .create();
                                }
                                return null;
                        }
                }
```





كيف يعمل المثال السابق ؟

في البداية حتى يمكننا أن نظهر المربع أو النموذج الحواري لابد أن نكتب هذا الإجراء الفرعي (الإجراء الفرعي عبارة عن برنامج صغير ، نكتبه لتنفيذ مهمة برمجية محددة) .

```
@Override
protected Dialog onCreateDialog(int id) {
//...المقصود هنا هو كافة الأوامر البرمجية السابقة التوضيح
}
```

هذا الإجراء السابق يتم استدعائه وتنفيذه حينما يتم إصدار الأمر ShowDialog(0) وهي المنطقة التالية من الكود .

```
Button btn = (Button) findViewById(R.id.btn_dialog);
btn.setOnClickListener(new View.OnClickListener() {
public void onClick(View v) {
showDialog(0);
}
});
```

الإجراء onCreateDialog يتم إدارة عملية تنفيذه من خلال النشاط الرئيسي للتطبيق و هو MainActivity ، حينما يصدر الأمر showDialog يرسل متغير بقيمة صفر إلى النشاط الرئيسي الذي يقوم بدوره بتنفيذ الإجراء الخاص بإظهار المربع الحواري ، هذا المربع الحواري يحتاج لكي يظهر إلى إستخدام أحد المكتبات الخاصة بلغة الجافا ، هذه المكتبة قمنا بإستيرادها من خلال الأمر import android.app.AlertDialog ، ويتبع ذلك إستيراد بعض المكتبات الخاصة بإظهار محتوى المربع الحواري مثل المكتبتين التاليتين import android.app.Dialog و import android.content.DialogInterface كذلك قمنا بتحيد بعض الخصائص للمربع الحواري مثل أيقونة المربع وعنون المربع وأزرار المربع وهما Ok, Cancel .

التعامل مع نافذة شريط الحالة Displaying a Progress Dialog .

سوف نقوم الآن باستخدام نفس المثال السابق ولكن لغرض آخر ، وهو شاشة أو نافذة شريط الحالة التي توضح لنا تطور حالة معينة من خلال شريط يظهر لنا ما قد مضى وما هو متبقي ، لن نغير أو نعدل أي شئ في ملف main.xml ولكننا سوف نطور التطبيق والكود المكتوب في الملف Activity.java بحيث يحتوي على تعديل أو إضافة برمجية جديدة مع المزيد من استدعاء للمكتبات التي تخدم الغرض الموضح في المثال . والآن نقوم بتعديل الكود ليصبح بنفس الشكل التالي

```
package net.learn2develop.Dialog;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.Dialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import android.app.ProgressDialog;
import android.os.Handler;
import android.os.Message;

public class DialogActivity extends Activity {

      CharSequence[] items = { "Google", "Apple", "Microsoft" };
      boolean[] itemsChecked = new boolean [items.length];

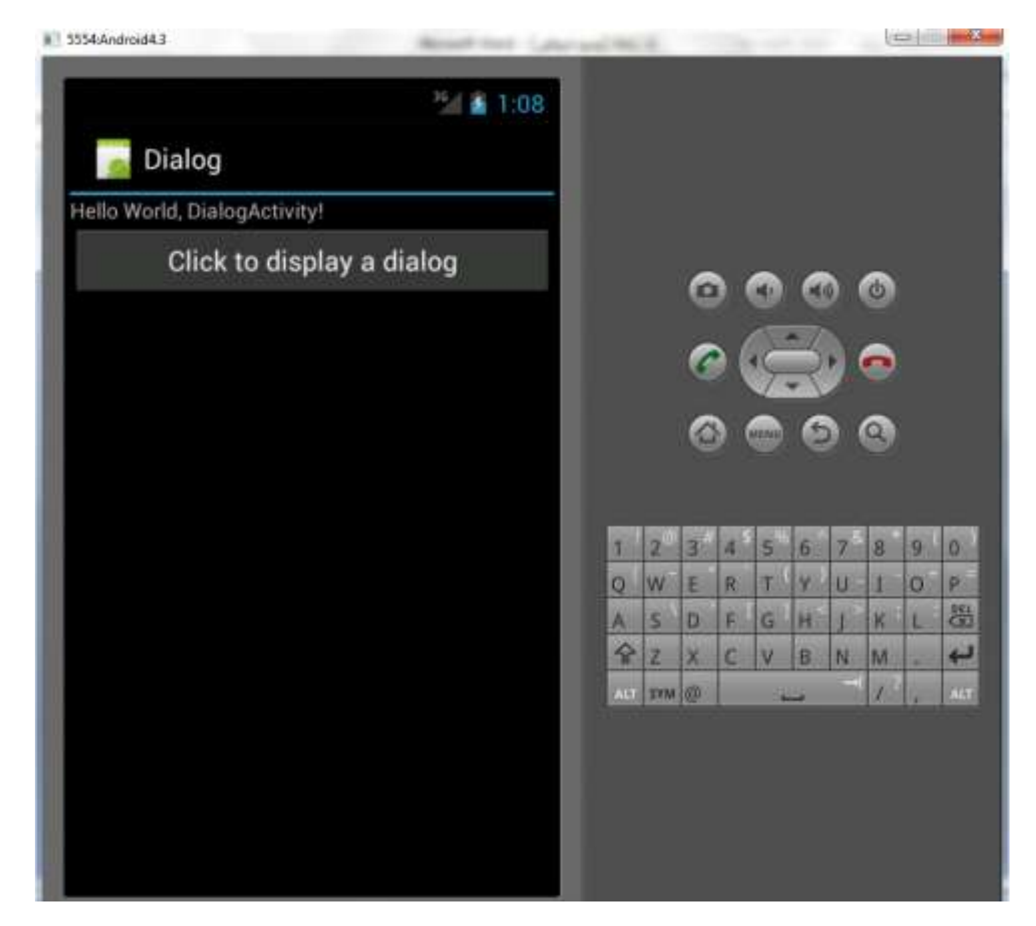
      private ProgressDialog _progressDialog;
      private int _progress = 0;
      private Handler _progressHandler;

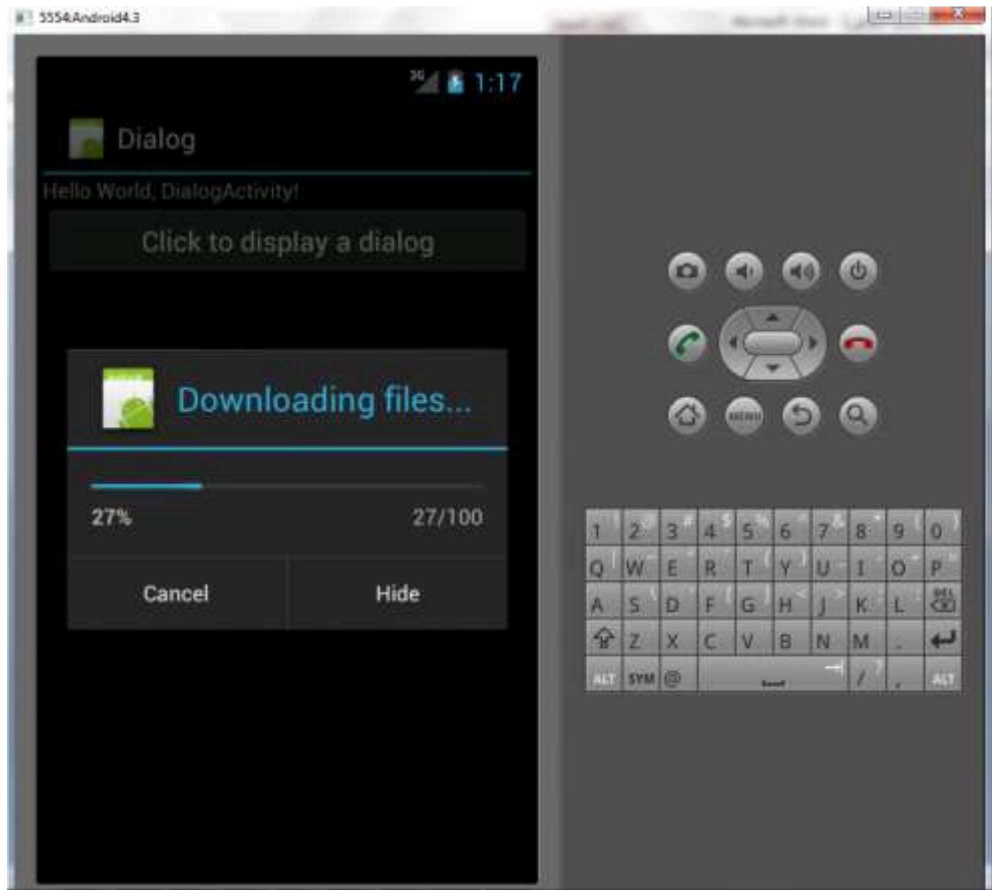
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        Button btn = (Button) findViewById(R.id.btn_dialog);
        btn.setOnClickListener(new View.OnClickListener() {
              public void onClick(View v) {
                      showDialog(1);
                      _progress = 0;
                      _progressDialog.setProgress(0);
                      _progressHandler.sendEmptyMessage(0);
                      }
        });
        _progressHandler = new Handler() {
              public void handleMessage(Message msg) {
                      super.handleMessage(msg);
                      if (_progress >= 100) {
                      _progressDialog.dismiss();
                      } else {
                      _progress++;
                      _progressDialog.incrementProgressBy(1);
                      _progressHandler.sendEmptyMessageDelayed(0, 100);
                      }
              }
        };
    }

    @Override
    protected Dialog onCreateDialog(int id) {
```

```
        switch (id) {
        case 0:
        return new AlertDialog.Builder(this)
        .setIcon(R.drawable.ic_launcher)
        .setTitle("This is a dialog with some simple text...")
        .setPositiveButton("OK", new
        DialogInterface.OnClickListener() {
              public void onClick(DialogInterface dialog,
              int whichButton)
        {
                Toast.makeText(getBaseContext(),"OK clicked!", Toast.LENGTH_SHORT).show();
        }
        })
        .setNegativeButton("Cancel", new
        DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog,
        int whichButton)
        {
          Toast.makeText(getBaseContext(),"Cancel clicked!", Toast.LENGTH_SHORT).show();
        }
        })
        .setMultiChoiceItems(items, itemsChecked, new
        DialogInterface.OnMultiChoiceClickListener() {
              @Override
              public void onClick(DialogInterface dialog, int which,
              boolean isChecked) {
              Toast.makeText(getBaseContext(),
              items[which] + (isChecked ? " checked!":
              " unchecked!"),
              Toast.LENGTH_SHORT).show();
              }
        }
        )
        .create();
        case 1:
          _progressDialog = new ProgressDialog(this);
          _progressDialog.setIcon(R.drawable.ic_launcher);
          _progressDialog.setTitle("Downloading files...");
          _progressDialog.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
          _progressDialog.setButton(DialogInterface.BUTTON_POSITIVE, "Hide", new
          DialogInterface.OnClickListener() {
          public void onClick(DialogInterface dialog,
          int whichButton)
          {
          Toast.makeText(getBaseContext(),
          "Hide clicked!", Toast.LENGTH_SHORT).show();
          }
          });
          _progressDialog.setButton(DialogInterface.BUTTON_NEGATIVE, "Cancel", new
          DialogInterface.OnClickListener() {
          public void onClick(DialogInterface dialog,
          int whichButton)
          {
          Toast.makeText(getBaseContext(),
          "Cancel clicked!", Toast.LENGTH_SHORT).show();
          }
          });
          return _progressDialog;
        }
        return null;
    }
}
```





كيف يعمل المثال السابق ؟

لكي نقوم بإظهار شريط الحالة Progress Dialog لابد أن نكتب الكود اللازم لاستخدام الوحدة البرمجية Progress Dialog Class الخاص به ، مع تحديد بعض الخصائص مثل الأيقونة والعنوان وشكل الشريط .

```
_progressDialog = new ProgressDialog(this);
_progressDialog.setIcon(R.drawable.icon);
_progressDialog.setTitle("Downloading files...");
_progressDialog.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
```

بعد ذلك قمنا بكتابة أوامر تعرض زرين أمر Buttons في نافذة شريط التمرير ، مع إعطاء كل زر الخصائص التي تعنيه .

```
_progressDialog.setButton(DialogInterface.BUTTON_POSITIVE, "Hide", new
DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog,
int whichButton)
{
Toast.makeText(getBaseContext(),
"Hide clicked!", Toast.LENGTH_SHORT).show();
}
});

_progressDialog.setButton(DialogInterface.BUTTON_NEGATIVE, "Cancel", new
DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog,
int whichButton)
{
Toast.makeText(getBaseContext(),
"Cancel clicked!", Toast.LENGTH_SHORT).show();
}
});
return _progressDialog;
}
```

لإظهار حالة شريط الحالة لابد وأن نتعامل مع عدة متغيرات أساسية وهي موضحة كمايلي .

```
_progress = 0;
_progressDialog.setProgress(0);
_progressHandler.sendEmptyMessage(0);
```

يعمل العداد في الخلفية على زيادة قيمة _prograess مع كل ١٠٠ ملي ثانية بقيمة زيادة واحد صحيح ، من خلال الأمر التالي ، وإذا وصلت القيمة لـ ١٠٠ يتم إخفاء شريط الحالة وإبطال تشغيله .

```
_progressHandler = new Handler() {
   public void handleMessage(Message msg) {
       super.handleMessage(msg);
       if (_progress >= 100) {
             _progressDialog.dismiss();
             } else {
                   _progress++;
                   _progressDialog.incrementProgressBy(1);
                   _progressHandler.sendEmptyMessageDelayed(0, 100);
             }
        }
};
```

ربط الأنشطة بالوحدات البرمجية linking Activities using intents .

في تطبيقات أندرويد من الممكن أن يكون عدد الأنشطة هو صفر أو أكثر ، وحينما يمتلك مشروعك أكثر من نشاط Activity فأنت تحتاج إلى التنقل بين هذه الأنشطة ، ولتستطيع فعل ذلك ينبغي عليك أن تفهم كيفية الربط بين الأنشطة المختلفة باستخدام intents . والأن سوف نقوم بإنشاء تطبيق جديد ونعطه الاسم linking Activities using intents ، ثم نحدد خاصية الـ Package Name بأنها تساوي net.learn2develop.laui وسوف نتبع التالي .

١ – قم بتعديل ملف AndroidMainfest.xml بحيث يصبح بالشكل التالي .

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.learn2develop.laui"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="15" />

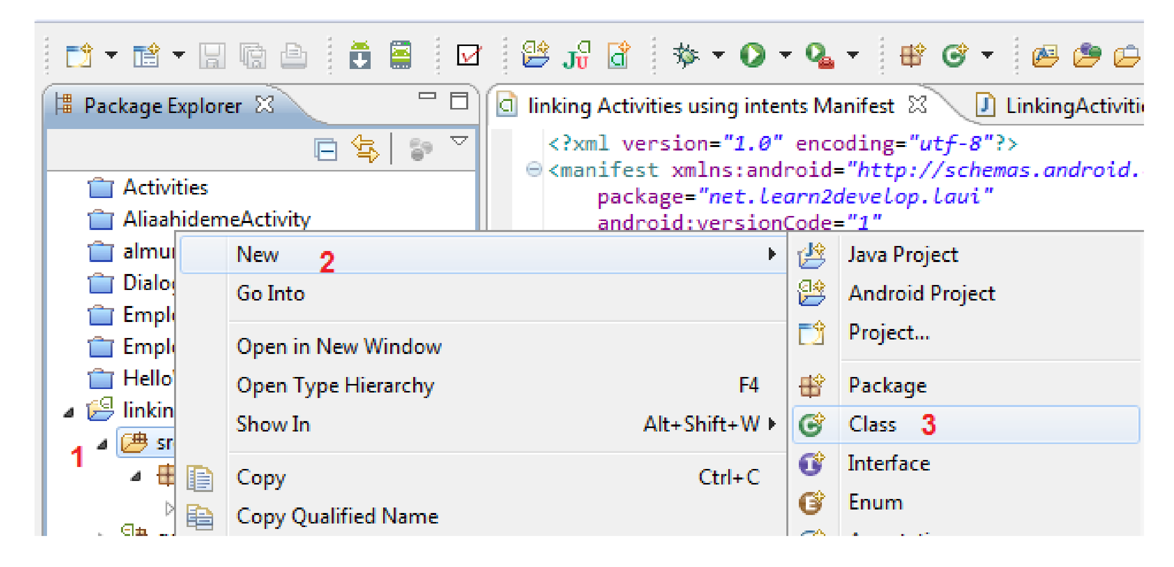
    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".LinkingActivitiesusingintentsActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

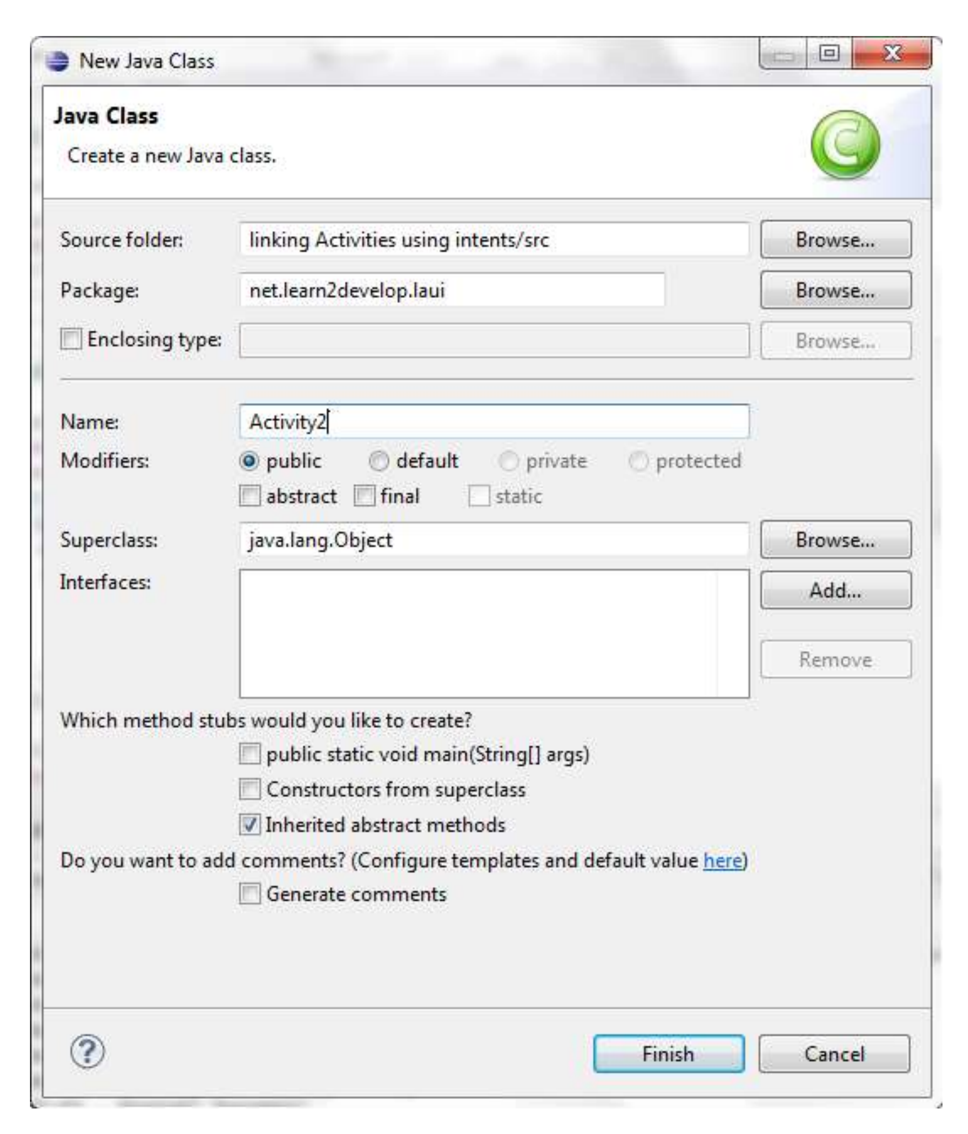
        <activity android:name=".Activity2"
             android:label="Activity 2">
             <intent-filter>
             <action android:name="net.learn2develop.ACTIVITY2" />
             <category android:name="android.intent.category.DEFAULT" />
             </intent-filter>
             </activity>
    </application>

</manifest>
```

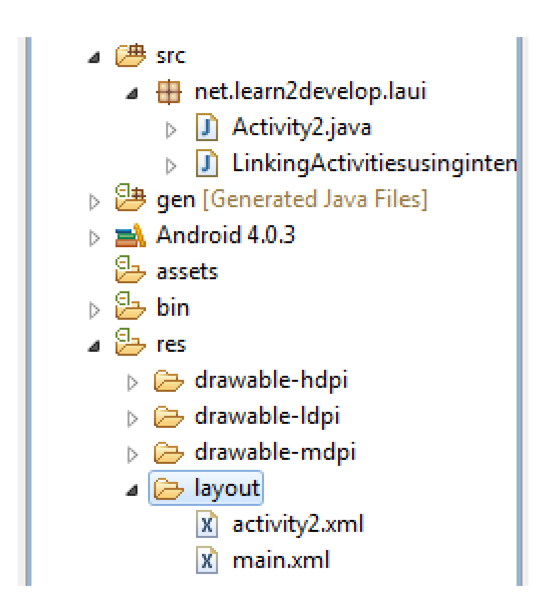
٢ – إضغط الزر الأيمن للماوس على المجلد src ثم أضف ملف class جديد للمشروع كما في الصورة التالية .



٣ – أضف ملف class جديد باسم Activity2 كما في الصورة التالية .



٤ – قم بنسخ الملف main.xml الموجود بمجلد layout وألصقه باسم activity2.xml ، بهذا الشكل لابد وأن يكون لديك نفس الملفات الموضحة بالصورة التالية .



٥ – قم بتعديل ملف activity2.xml بحيث يصبح كما في الكود التالي .

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
```

```
    <TextView
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="This is Activity 2!"
/>

</LinearLayout>
```

٦ – في ملف Activity2.java قم بتعديل الكود ليصبح كما في الكود التالي .

```
package net.learn2develop.laui;

import android.app.Activity;
import android.os.Bundle;

public class Activity2 extends Activity {
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity2);
}
}
```

٧ – قم بتعديل ملف MainActivity.java بحيث يصبح كالكود التالي .

```
package net.learn2develop.laui;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;
import android.view.Window;

import android.view.KeyEvent;
import android.content.Intent;

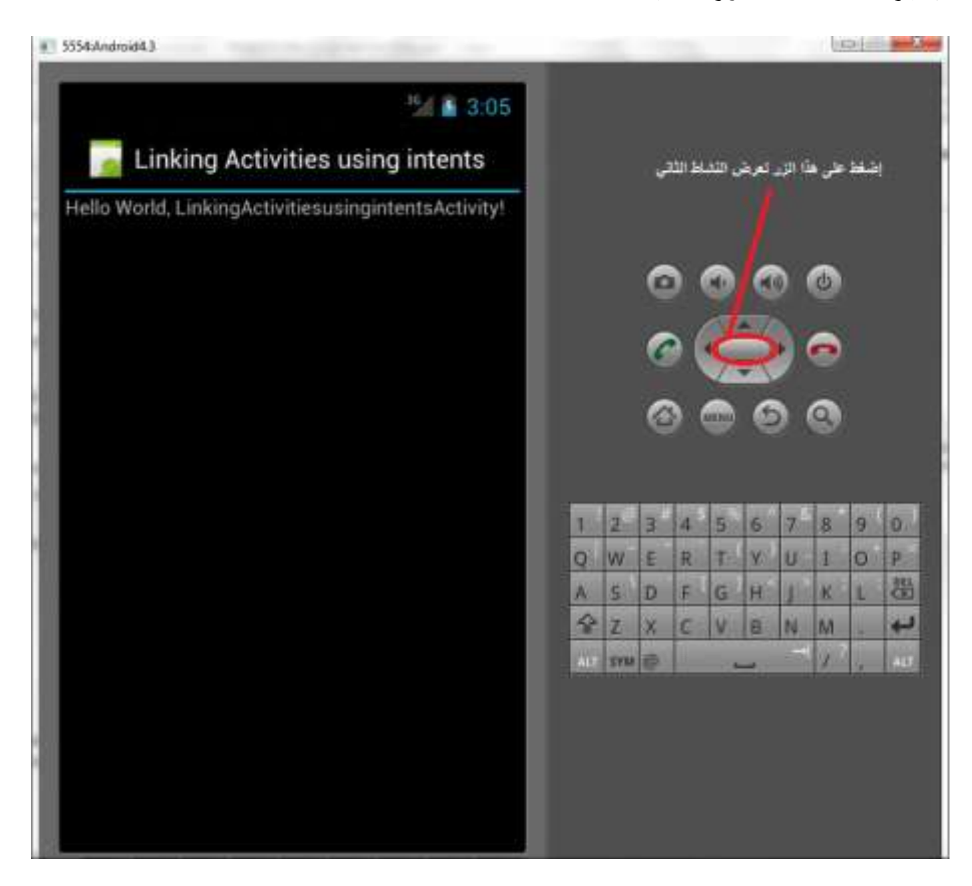
public class LinkingActivitiesusingintentsActivity extends Activity {

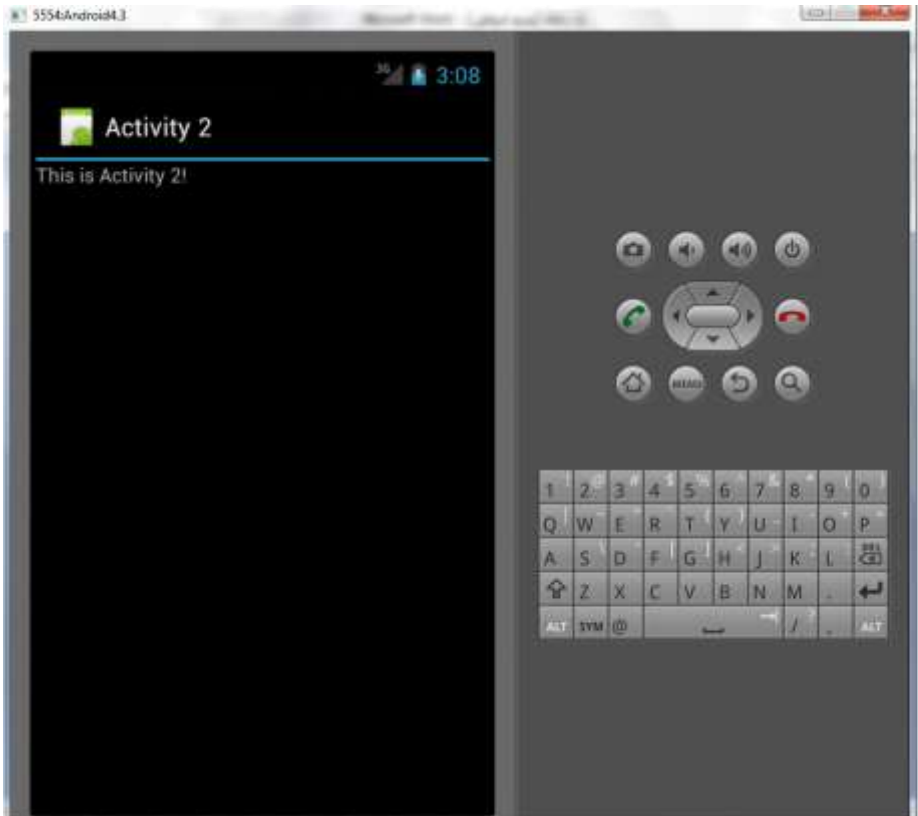
      String tag = "Events";

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Log.d(tag, "In the onCreate() event");
    }
    public boolean onKeyDown(int keyCode, KeyEvent event)
    {
      if (keyCode == KeyEvent.KEYCODE_DPAD_CENTER)
      {
            startActivity(new Intent("net.learn2develop.ACTIVITY2"));
      }
    return false;
    }
```

```
}
```

قم بتشغيل التطبيق وسوف يظهر أمامك كما بالصورة التالية .





حل مشكلة فرز تجمعات الوحدات البرمجية Resolving Intent Filter Collision .

قد يصادفك مشروع معين ويحتاج منك إلى إنشاء عدة أنشطة Activity ، لكل نشاط اسم خاص به بالطبع ، ولكن كلاهما يعمل من خلال نفس الشاشة أو التصميم ، مثلاً تريد أن تصمم نافذة لتسجيل بيانات معينة عن أحد المنتجات ، ولكنك تحتاج قبل ذلك إلى معرفة هل سوف يتم تسجيل البيانات كبيانات منتج مطلوب شرائه أم بيانات منتج معروض للبيع ؟ ، لذا فسوف نوضح هنا كيف يقوم نظام التشغيل بعرض سؤال ليستفهم منك عن أياً من الأنشطة تريد التعامل معه ، كل ما سوف نقوم بعمله هنا هو تعديل المثال السابق بحيث يتغير ملف AndroidMainfest.xml ليصبح كما بالمثال التالي .

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.learn2develop.laui"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="15" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".LinkingActivitiesusingintentsActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity android:name=".Activity2"
             android:label="Activity 2">
                   <intent-filter>
                   <action android:name="net.learn2develop.ACTIVITY2" />
                   <category android:name="android.intent.category.DEFAULT" />
                   </intent-filter>
             </activity>

        <activity android:name=".Activity3"
             android:label="Activity 3">
                   <intent-filter>
                   <action android:name="net.learn2develop.ACTIVITY2" />
                   <category android:name="android.intent.category.DEFAULT" />
                   </intent-filter>
             </activity>

    </application>

</manifest>
```



أسئلة الفصل الرابع

س١ : أكتب تطبيق يسأل المستخدم عن اختياره من بين عدة أختيارات ؟

س٢ : أكتب تطبيق يعرض أثنين من الأنشطة **Activity** ويقوم بالتبديل بينهما ؟